```python
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

df = pd.read_csv("mymoviedb.csv", lineterminator = "\n")
df
```

```
     Release_Date                                    Title  \
0      2021-12-15                  Spider-Man: No Way Home
1      2022-03-01                               The Batman
2      2022-02-25                                  No Exit
3      2021-11-24                                  Encanto
4      2021-12-22                           The King's Man
...           ...                                      ...
9822   1973-10-15                                 Badlands
9823   2020-10-01                         Violent Delights
9824   2016-05-06                             The Offering
9825   2021-03-31  The United States vs. Billie Holiday
9826   1984-09-23                                  Threads

                                               Overview  Popularity  \
0     Peter Parker is unmasked and no longer able to...    5083.954
1     In his second year of fighting crime, Batman u...    3827.658
2     Stranded at a rest stop in the mountains durin...    2618.087
3     The tale of an extraordinary family, the Madri...    2402.201
4     As a collection of history's worst tyrants and...    1895.511
...                                                 ...         ...
9822  A dramatization of the Starkweather-Fugate kil...      13.357
9823  A female vampire falls in love with a man she ...      13.356
9824  When young and successful reporter Jamie finds...      13.355
9825  Billie Holiday spent much of her career being ...      13.354
9826  Documentary style account of a nuclear holocau...      13.354

      Vote_Count  Vote_Average Original_Language  \
0           8940           8.3                en
1           1151           8.1                en
2            122           6.3                en
3           5076           7.7                en
4           1793           7.0                en
...          ...           ...               ...
9822         896           7.6                en
9823           8           3.5                es
9824          94           5.0                en
9825         152           6.7                en
9826         186           7.8                en

                                  Genre  \
0     Action, Adventure, Science Fiction
1               Crime, Mystery, Thriller
```

```
2                                Thriller
3     Animation, Comedy, Family, Fantasy
4       Action, Adventure, Thriller, War
...                                   ...
9822                        Drama, Crime
9823                              Horror
9824             Mystery, Thriller, Horror
9825                 Music, Drama, History
9826           War, Drama, Science Fiction

                                             Poster_Url
0     https://image.tmdb.org/t/p/original/1g0dhYtq4i...
1     https://image.tmdb.org/t/p/original/74xTEgt7R3...
2     https://image.tmdb.org/t/p/original/vDHsLnOWKl...
3     https://image.tmdb.org/t/p/original/4j0PNHkMr5...
4     https://image.tmdb.org/t/p/original/aq4Pwv5Xeu...
...                                                 ...
9822  https://image.tmdb.org/t/p/original/z81rBzHNgi...
9823  https://image.tmdb.org/t/p/original/4b6HY7rud6...
9824  https://image.tmdb.org/t/p/original/h4uMM1wOhz...
9825  https://image.tmdb.org/t/p/original/vEzkxuE2sJ...
9826  https://image.tmdb.org/t/p/original/lBhU4U9Eeh...

[9827 rows x 9 columns]
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9827 entries, 0 to 9826
Data columns (total 9 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Release_Date       9827 non-null   object
 1   Title              9827 non-null   object
 2   Overview           9827 non-null   object
 3   Popularity         9827 non-null   float64
 4   Vote_Count         9827 non-null   int64
 5   Vote_Average       9827 non-null   float64
 6   Original_Language  9827 non-null   object
 7   Genre              9827 non-null   object
 8   Poster_Url         9827 non-null   object
dtypes: float64(2), int64(1), object(6)
memory usage: 691.1+ KB
```

```
df.duplicated().sum()
```

```
0
```

```
df.describe()
```

```
        Popularity    Vote_Count  Vote_Average
count  9827.000000   9827.000000   9827.000000
mean     40.326088   1392.805536      6.439534
std     108.873998   2611.206907      1.129759
min      13.354000      0.000000      0.000000
25%      16.128500    146.000000      5.900000
50%      21.199000    444.000000      6.500000
75%      35.191500   1376.000000      7.100000
max    5083.954000  31077.000000     10.000000
```

##Exploration Summary • we have a dataframe consisting of 9827 rows and 9 columns. • our dataset looks a bit tidy with no NaNs nor duplicated values. • Release_Date column needs to be casted into date time and to extract only the • Overview, Original_Languege and Poster-Url wouldn't be so useful during analys • there is noticable outliers in Popularity column • Vote_Average bettter be categorised for proper analysis. • Genre column has comma saperated values and white spaces that needs to be hand

```
df.isna().sum()
```

```
Release_Date         0
Title                0
Overview             0
Popularity           0
Vote_Count           0
Vote_Average         0
Original_Language    0
Genre                0
Poster_Url           0
dtype: int64
```

```
df["Release_Date"] = pd.to_datetime(df["Release_Date"])

df["Release_Date"] = df["Release_Date"].dt.year

df["Release_Date"]
```

```
0       2021
1       2022
2       2022
3       2021
4       2021
        ...
9822    1973
9823    2020
9824    2016
9825    2021
9826    1984
Name: Release_Date, Length: 9827, dtype: int32
```

```
cols = ["Overview","Original_Language","Poster_Url"]
```

```
df.drop(cols , axis = 1 , inplace = True)
df

      Release_Date                                    Title
Popularity  \
0             2021                 Spider-Man: No Way Home    5083.954

1             2022                              The Batman    3827.658

2             2022                                 No Exit    2618.087

3             2021                                 Encanto    2402.201

4             2021                          The King's Man    1895.511

...            ...                                     ...         ...

9822          1973                                Badlands      13.357

9823          2020                        Violent Delights      13.356

9824          2016                             The Offering      13.355

9825          2021  The United States vs. Billie Holiday      13.354

9826          1984                                 Threads      13.354


      Vote_Count  Vote_Average                                   Genre
0           8940           8.3  Action, Adventure, Science Fiction
1           1151           8.1            Crime, Mystery, Thriller
2            122           6.3                            Thriller
3           5076           7.7  Animation, Comedy, Family, Fantasy
4           1793           7.0    Action, Adventure, Thriller, War
...          ...           ...                                 ...
9822         896           7.6                       Drama, Crime
9823           8           3.5                              Horror
9824          94           5.0           Mystery, Thriller, Horror
9825         152           6.7             Music, Drama, History
9826         186           7.8        War, Drama, Science Fiction

[9827 rows x 6 columns]

def categorize_col (df, col, labels):
    edges = [df[col].describe()['min'],
             df[col].describe()['25%'],
             df[col].describe()['50%'],
             df[col].describe()['75%'],
             df[col].describe()['max']]

    df[col] = pd.cut(df[col], edges, labels = labels,
duplicates='drop')
    return df

labels = ['not_popular', 'below_avg', 'average', 'popular']

categorize_col(df, 'Vote_Average', labels)

df['Vote_Average'].unique()

['popular', 'below_avg', 'average', 'not_popular', NaN]
Categories (4, object): ['not_popular' < 'below_avg' < 'average' <
'popular']

df.head()

   Release_Date                    Title  Popularity  Vote_Count
Vote_Average  \
0          2021  Spider-Man: No Way Home    5083.954        8940
popular
1          2022               The Batman    3827.658        1151
popular
2          2022                  No Exit    2618.087         122
below_avg
3          2021                  Encanto    2402.201        5076
popular
4          2021           The King's Man    1895.511        1793
average

                                Genre
0  Action, Adventure, Science Fiction
1            Crime, Mystery, Thriller
2                            Thriller
3  Animation, Comedy, Family, Fantasy
4    Action, Adventure, Thriller, War

df["Vote_Average"].value_counts()

Vote_Average
not_popular    2467
popular        2450
average        2412
below_avg      2398
Name: count, dtype: int64

df["Genre"]= df["Genre"].str.split(", ")
df = df.explode('Genre').reset_index(drop=True)
df.head()

   Release_Date                    Title  Popularity  Vote_Count
Vote_Average  \
0          2021  Spider-Man: No Way Home    5083.954        8940
```

```
popular
1         2021  Spider-Man: No Way Home    5083.954        8940
popular
2         2021  Spider-Man: No Way Home    5083.954        8940
popular
3         2022               The Batman    3827.658        1151
popular
4         2022               The Batman    3827.658        1151
popular

             Genre
0           Action
1        Adventure
2  Science Fiction
3            Crime
4          Mystery
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 25793 entries, 0 to 25792
Data columns (total 6 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Release_Date  25793 non-null  int32
 1   Title         25793 non-null  object
 2   Popularity    25793 non-null  float64
 3   Vote_Count    25793 non-null  int64
 4   Vote_Average  25552 non-null  category
 5   Genre         25793 non-null  object
dtypes: category(1), float64(1), int32(1), int64(1), object(2)
memory usage: 932.3+ KB
```

```
df.nunique()
```

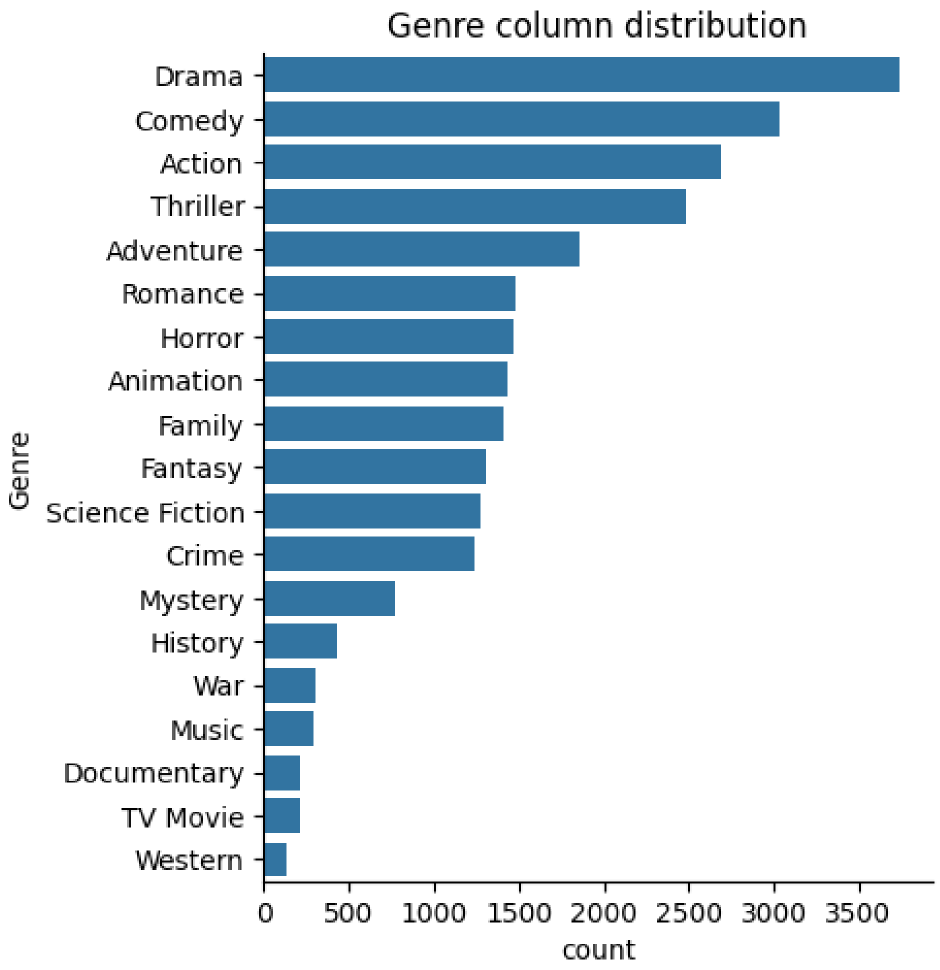
```
Release_Date     102
Title           9513
Popularity      8160
Vote_Count      3266
Vote_Average       4
Genre             19
dtype: int64
```

What is the most frequent genre in the dataset?

```
df["Genre"].describe()
```

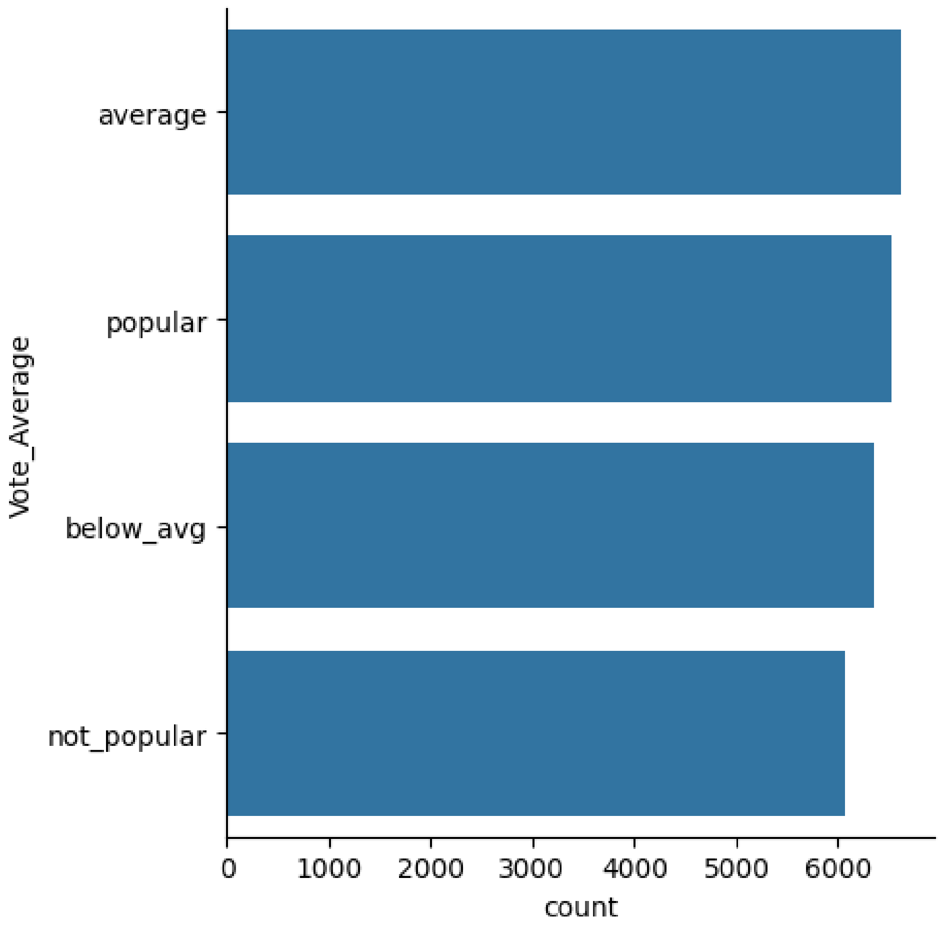
```
count     25793
unique       19
top       Drama
```

```
freq         3744
Name: Genre, dtype: object

sns.catplot(y = 'Genre', data = df, kind = 'count', order =
df['Genre'].value_counts().index)
plt.title('Genre column distribution')

Text(0.5, 1.0, 'Genre column distribution')
```



What has highest votes in vote avg.column?

```
sns.catplot(y="Vote_Average", data= df, kind= "count", order =
df["Vote_Average"].value_counts().index)

<seaborn.axisgrid.FacetGrid at 0x146d2f23fe0>
```

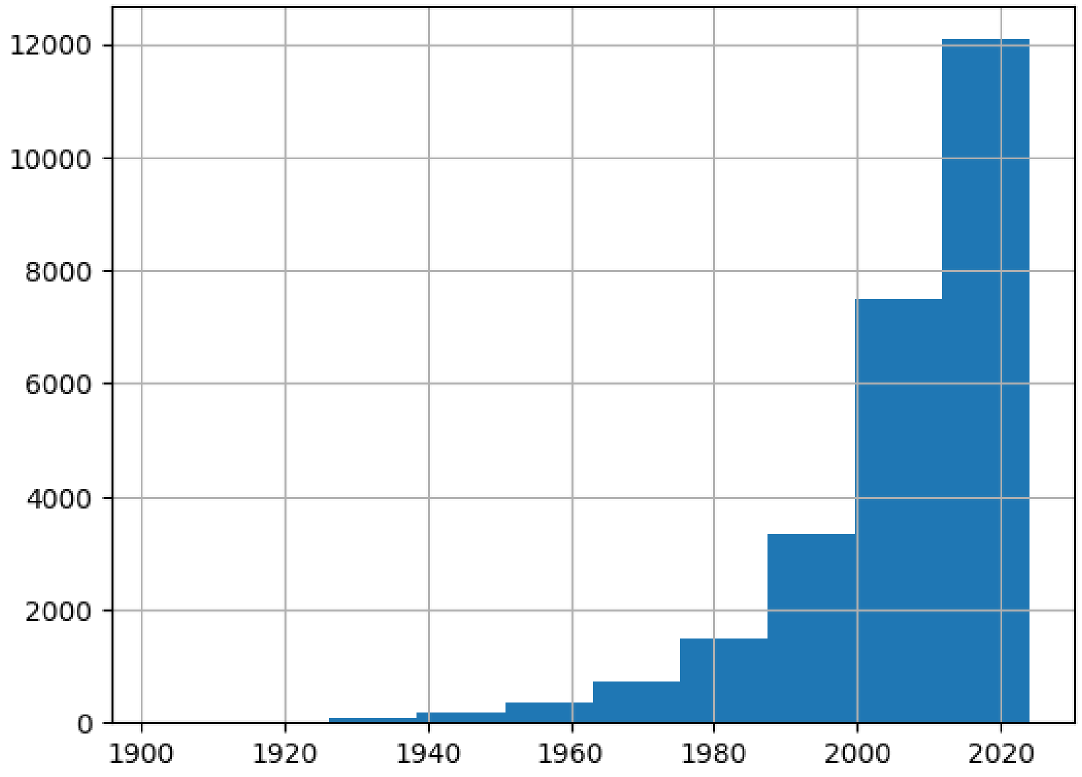


```
df[df["Popularity"] == df["Popularity"].max()]

   Release_Date                    Title  Popularity  Vote_Count
Vote_Average  \
0          2021  Spider-Man: No Way Home    5083.954        8940
popular
1          2021  Spider-Man: No Way Home    5083.954        8940
popular
2          2021  Spider-Man: No Way Home    5083.954        8940
popular

             Genre
0           Action
1        Adventure
2  Science Fiction

df[df["Popularity"] == df["Popularity"].min()]

       Release_Date                                 Title  Popularity
\
25787          2021  The United States vs. Billie Holiday      13.354
```

```
25788            2021  The United States vs. Billie Holiday      13.354

25789            2021  The United States vs. Billie Holiday      13.354

25790            1984                                 Threads      13.354

25791            1984                                 Threads      13.354

25792            1984                                 Threads      13.354


       Vote_Count Vote_Average            Genre
25787         152      average            Music
25788         152      average            Drama
25789         152      average          History
25790         186      popular              War
25791         186      popular            Drama
25792         186      popular  Science Fiction
```

```python
df["Release_Date"].hist()
```

```
<Axes: >
```



12000
10000
8000
6000
4000
2000
0
1900
1920
1940
1960
1980
2000
2020